

Deep Learning Strategies for Industrial Surface Defect Detection Systems

Dominik Martin
Karlsruhe Institute
of Technology (KIT)
dominik.martin@kit.edu

Simon Heinzel
Karlsruhe Institute
of Technology (KIT)
simon.heinzel@alumni.kit.edu

Johannes Kunze von Bischhoffshausen
Trelleborg Sealing Solutions
Germany GmbH
johannes.kunze@trelleborg.com

Niklas Kühl
Karlsruhe Institute
of Technology (KIT)
niklas.kuehl@kit.edu

Abstract

Deep learning methods have proven to outperform traditional computer vision methods in various areas of image processing. However, the application of deep learning in industrial surface defect detection systems is challenging due to the insufficient amount of training data, the expensive data generation process, the small size, and the rare occurrence of surface defects. From literature and a polymer products manufacturing use case, we identify design requirements which reflect the aforementioned challenges. Addressing these, we conceptualize design principles and features informed by deep learning research. Finally, we instantiate and evaluate the gained design knowledge in the form of actionable guidelines and strategies based on an industrial surface defect detection use case. This article, therefore, contributes to academia as well as practice by (1) systematically identifying challenges for the industrial application of deep learning-based surface defect detection, (2) strategies to overcome these, and (3) an experimental case study assessing the strategies' applicability and usefulness.

In the past, these systems relied on traditional computer vision methods, which addressed at least some of the issues of manual visual inspection [4]. However, with the Industry 4.0 paradigm, which aims to increase automation of traditional manufacturing processes through digitization, the trend is moving towards the generalization of the production line, where rapid adaptation to a new product is required [5]. Traditional computer vision methods are unable to provide such flexibility. They rely on a two-step process of extracting handcrafted features and training an appropriate classifier. The critical step in this process lies in the extraction of robust handcrafted feature representations for the specific problem at hand [6]. This step leads to lengthy development cycles [2] and requires a high level of human expertise [7]. A solution that allows for improved flexibility and reduced engineering efforts can be found in deep learning methods. Deep learning methods learn the relevant features directly from the raw data, eliminating the need for handcrafted feature representations. In recent years, these methods have reached and even exceeded human-level performance on image-related tasks such as image classification [8].

1. Introduction

Quality control is an essential process in the manufacturing industry [1]. As part of quality management, it ensures the quality of manufactured products. In this process, the visual inspection of finished products plays an important role [2]. Typically, this task is carried out manually, and workers are trained to identify complex surface defects [3]. However, manual visual inspection is monotonous, laborious, fatiguing, subjective, lacking in good reproducibility, too slow in many cases, and costly. As a result, automated visual inspection systems have spread in the industry since the 1970s. The main benefits of such systems include impartial and reproducible inspection results, complete and detailed documentation, faster inspection rates, and lower costs [3, 2].

However, deep learning methods are still rarely applied in automated visual inspection systems due to several reasons [9]. The available datasets are usually too small to train deep neural networks [10, 6, 11, 7, 1, 12, 13] and the generation of such datasets is expensive due to the intensive manual work required for labeling the data [7, 12]. Additionally, surface defects can be extremely small, making their detection even more challenging [11, 14]. The black-box nature of deep neural networks also makes it difficult for human domain experts to understand what the network considers a defect [15].

Against this background, we contribute to the information systems (IS) literature by investigating suitable strategies that enable the successful application of deep learning methods in industrial surface defect detection systems (SDDS). More specifically, we aim to

answer the following research questions:

- RQ1:** Which challenges exist for deep learning methods in industrial SDDS, and which design requirements can be derived from these challenges?
- RQ2:** Which deep learning strategies in the form of design principles and design features address these design requirements and are suitable for industrial SDDS?
- RQ3:** Which strategies achieve the best performance in industrial SDDS?

2. Research Design

To address the research questions raised, we follow the Design Science Research (DSR) paradigm [16, 17]. Overall, we base our research on the three cycle view proposed by Hevner [18], which ensures practical applicability on the one hand and rigorous construction and evaluation of innovative artifacts on the other. Thus, we aim to create artifacts that solve the problems of a specific application domain (relevance cycle) while drawing on applicable knowledge from theory (rigor cycle). In this particular research, we contribute to the application domain of surface defect detection in the manufacturing industry and base our artifact construction on literature from the field of deep learning.

Our specific approach is based on the DSR process model presented by Peffers et al. [19] and consists of six subsequent steps. Figure 1 illustrates these steps, the resulting outputs, and the corresponding research activities. First, we define the research problem by identifying domain-specific challenges for deep learning methods in relevant literature as well as through exploratory focus groups in a case company [20]. In several focus group sessions conducted, seven experts from different areas such as operations, quality control and data science were involved. From the challenges identified, we derive *design requirements* (DR), which represent generic requirements that should be met by any artifact aiming to solve these problems [21]. This step corresponds to the relevance cycle and addresses *RQ1*. Second, we define the objectives of a solution by inferring *design principles* (DP) from the design requirements. Design principles are generic capabilities of an artifact through which the design requirements are addressed [21]. We base the design principles on relevant literature from the field of deep learning; hence this step corresponds to the rigor cycle. In the third step, we derive *design features* (DF) that address

the design principles and conceptualize a framework of interrelated design requirements, design principles, and design features. Design features are specific capabilities of an artifact that fulfill and implement the design principles [21]. A design principle that is instantiated by a design feature can be understood as an explanation (design principle) of why a specified piece (design feature) leads to a predefined goal (design requirement). This step corresponds to the first design cycle and, together with the previous step, answers *RQ2*. Next, we validate the artifact proposed in the first design cycle and demonstrate its feasibility, applicability and usefulness [22] by instantiating it in the context of an exemplary surface defect detection use case. We conduct eight experiments leveraging strategies from the framework. In step five, we evaluate the deep learning models and draw conclusions about the different deep learning strategies. Steps four and five, thus, address *RQ3* and represent the second design cycle. Finally, we contribute to the body of knowledge by communicating the identified challenges, the created artifacts, and the evaluation results in the article at hand.

In summary, the first artifact is a *framework* of interrelated design requirements, principles, and features which captures suitable deep learning strategies for enabling industrial surface defect detection systems. The second artifact is an *instantiation* of the framework on an industrial use case in the field of visual inspection of engineered molded parts. In a series of experiments, we build different deep learning models leveraging strategies from the framework illustrating their feasibility, applicability and usefulness. Thus, this article aims to contribute design knowledge in the form of *operational principles/architectures* and a *situated implementation of an artifact* [23]. Hence, it makes a *level 2* (design cycle 1) and *level 1* (design cycle 2) contribution according to Gregor and Hevner [23]. The DSR knowledge contribution type represents an *exaptation*, since this article aims to extend known solutions to new problems [23].

3. Relevance Cycle: Design Requirements for Surface Defect Detection Systems

The relevance cycle aims to place the research in a contextual environment and provide requirements and acceptance criteria for the design science activities. Thus, we provide an overview of previous research on surface defect detection in the manufacturing industry to identify domain-specific challenges for deep learning methods. By leveraging insights from related literature as well as an exploratory case study with experts from industry, identified challenges are condensed into design

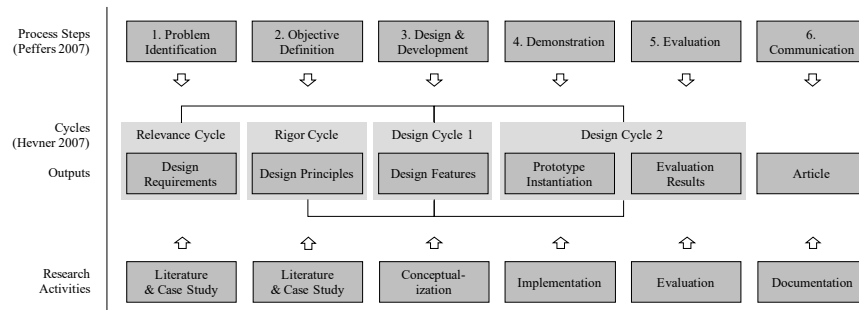


Figure 1: Overall research design based on Peffers et al. [19] and Hevner [18]

requirements.

3.1. Surface Defect Detection

The detection of surface defects using computer vision techniques has been widely studied in the literature. Surface defects are considered local anomalies in homogeneous textures like scratches, cracks, holes, etc. This includes a wide range of surface textures, including textile [24], wood [25], metal [26] and ceramic tiles [27]. The methods commonly leveraged can be divided into *traditional computer vision methods* and *deep learning methods*. Traditional computer vision methods are based on a two-step process of extracting handcrafted features and training an appropriate classifier such as an SVM or decision tree. The critical step in this process lies in the extraction of robust handcrafted feature representations for the specific problem at hand.

Xie [4] categorizes the methods used to extract these features into four different approaches: *Structural approaches* focus on texture elements and their spatial arrangement. They extract texture primitives such as simple line segments, individual pixels, or regions with uniform gray-levels and generate a dynamic texture model by applying some spatial placement rules. Structural methods are usually applied to repetitive patterns such as textile [28], fabrics [29], and leather [30]. Popular structural approaches include primitive measurement [31], edge features [30], skeleton representation [28], and morphological operations [29]. *Statistical approaches* analyze the spatial distribution of pixel values. They work well on stochastic textures, such as ceramic tiles, castings, and woods. In this category, researchers use numerous statistics, such as histogram properties [32], co-occurrence matrices [33], local binary patterns [34], autocorrelation [35] and others. *Filter-based approaches* apply filters to detect features, such as edges, textures, and regions. They can be further divided into spatial domain

filtering [36], frequency domain filtering [37], and spatial-frequency domain filtering [38]. *Model-based approaches* construct representations of images by modeling multiple properties of the defects. In this category, researchers use fractal models [39], autoregressive models [40], and random field models [41].

Shortly after the introduction of AlexNet [42], deep learning methods began being applied more often to surface defect detection problems. The motivation arises from the difficulty that even domain experts struggle to design the right set of features to detect certain defects. Masci et al. [43] show that deep learning methods can significantly outperform traditional computer vision methods. They use a CNN consisting of five layers for the classification of steel defects and achieved excellent results. The work from Soukup and Huber-Mörk [10] shows that regularization methods like unsupervised layer-wise pre-training and data augmentation yield further performance improvements. Weimer et al. [6] evaluate several deep learning architectures with varying depths and widths of layers on a synthetic texture dataset. The work from Ren et al. [7] shows that using a pre-trained network improves the performance of deep learning methods. They also extend the problem of surface defect detection from image classification to image segmentation.

3.2. Design Requirements

However, especially the application of deep learning approaches opens up a number of previously inadequately explored challenges. One challenge, pointed out by several authors, is the particularly small size of the defects [11, 14]. Also in our selected industrial use case, the defects are so small that they are difficult to see with the naked eye. This makes it more difficult to detect the defects and capture them in a way that the defects are also visible in the images. Consequently, we derive the following design

requirement:

DR1: *Industrial surface defect detection systems should be able to detect very small defects.*

A second challenge lies in the rare occurrence of defects [7, 44, 13]. Datasets from manufacturing processes are often highly imbalanced due to the deliberately low probability of defect occurrences. Deep learning methods in general are designed to minimize the overall loss, which can result in paying more attention to the majority class and not properly learning the appearance of the minority class. Consequently, this issue has to be addressed appropriately:

DR2: *Industrial surface defect detection systems should be able to detect rarely occurring defects.*

A third challenge concerns the difficulty in understanding deep neural networks [15]. Deep neural networks are black-box networks, making them difficult to understand or interpret [45]. The quality inspectors in our use case also emphasize the importance of trusting deep learning methods because their model decisions as such are untraceable; thus:

DR3: *The decisions of industrial surface defect detection systems should be explainable.*

A fourth challenge is the insufficient amount of training data. Several authors point out that the size of datasets is usually too small to train deep neural networks and that the training is prone to overfitting [10, 6, 11, 7, 1, 12, 13]. A fifth challenge is the expensive data generation process. A series of recent studies remarks that the acquisition of images and especially the labeling of images is costly due to the required expert knowledge and intensive manual work [7, 12]. Consequently, we derive the fourth design requirement:

DR4: *Industrial surface defect detection systems should be able to learn from small amounts of training data.*

4. Rigor Cycle: Drawing on Deep Learning Theory

To address the design requirements derived in the previous section, we identify design principles by drawing on relevant literature as well as insights from domain experts in the field of visual inspection. Design principles are generic capabilities of an artifact through which the design requirements are addressed.

Since defects are often very small in relation to the dimensions of the examined part, they can only be captured appropriately by capturing multiple segments of the part rather than photographing the entire part at once. Consequently, we derive the first design principle, which addresses DR1:

DPI1: *Provide the system with segment-wise*

examination capabilities.

Shang et al. [44] remark that deep neural nets should be trained on balanced datasets to make more reliable predictions. Oversampling and undersampling are common techniques to adjust the class distribution of a dataset. Oversampling techniques oversample the minority class to create a balanced dataset, and undersampling strategies undersample the majority class to create a balanced dataset. Consequently, we derive the second design principle, which addresses DR2:

DP2: *Provide the system with data balancing functions.*

Several authors address the problem of surface defect detection as a binary classification problem [10, 11, 2]. They argue that an accurate per-image classification is often more important than an accurate localization of the defect. Others address the problem as a multi-class classification problem, where the model has to specify the defect type [43, 6, 46, 14, 12, 47]. Some authors argue that the precise localization of defects is crucial and address the problem as a segmentation problem [7, 1, 15]. Segmentation models output a visual localization of the defect in the form of a segmentation map, which provide higher information value compared to binary or even multi-class classification. However, there seems to be no consensus on how to address the problem of surface defect detection. Instead, the problem of surface defect detection is addressed according to the goals and priorities of the specific use case. Consequently, we derive the third design principle, which addresses DR3 and DR4:

DP3: *Provide the system with mechanisms to address the appropriate information needs of the users based on the objectives and priorities of the specific use case.*

Previous research shows that the use of pre-trained weights from large image datasets yields performance improvements over deep neural networks trained from scratch [7, 44, 47, 1]. Consequently, we derive the fourth design principle, which addresses DR4:

DP4: *Provide the system with knowledge transfer functions to utilize shared features from other models.*

Recent studies indicate that regularization methods prevent deep neural networks from overfitting and thus improve model performance. Popular regularization methods include dropout [14, 1, 13] and data augmentation [10, 6, 14, 12, 13]. Dropout is a technique that randomly drops units and their connections from the network during training [48]. This prevents the units from co-adapting too much. Data augmentation is a technique that increases the diversity of the training set by applying random but realistic transformations such as

flipping and rotation. Consequently, we derive the fifth design principle, which addresses DR4:

DP5: *Provide the system with regularization mechanisms to prevent the model from overfitting.*

5. Design Cycle 1: Strategies for Enabling Deep Learning-based SDDS

Building on the design principles presented in the previous section, this section derives design features capturing concrete instantiations in the specific context of industrial surface defect detection use cases. Figure 2 depicts an overview of the design features, design principles, and design requirements.

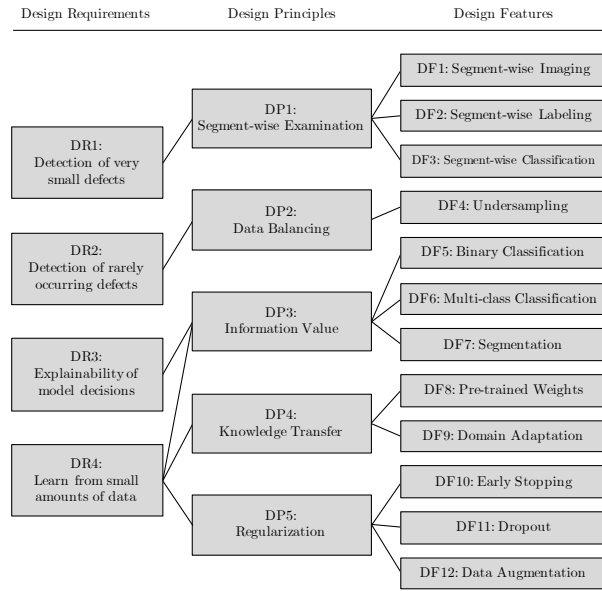


Figure 2: Framework of interrelated design requirements, design principles, and design features

Three design features implement the first design principle. First, we photograph a product in segments (DF1), second, we label each segment/image (DF2) and, third, we derive a product classification based on several individual segment classifications (DF3). The second design principle is addressed by undersampling the majority class to adjust the class distribution so that each class is equally represented (DF4). The third design principle requires defining the appropriate information value and is addressed by three different design features. The model can either output a binary classification (DF5), a multi-class classification (DF6), or a segmentation map (DF7). The fourth design principle postulates knowledge transfer and is implemented by using pre-trained weights (DF8) or a domain adaptation method (DF9). The pre-trained

weights come from a dataset like ImageNet or some other large image dataset. The fifth design principle is addressed by stopping the training of a model early when a monitored metric has stopped improving (DF10), randomly dropping units and their connections during training (DF11), and applying random transformations to the training data (DF12).

6. Design Cycle 2: Surface Defect Detection Prototype Instantiation

To evaluate the applicability and usefulness of the framework on an industrial use case, we conduct eight technical experiments in which we instantiate different deep learning strategies.

6.1. Industrial Use Case

We evaluate the proposed SDDS in a company, which manufactures so-called engineered molded parts, which are custom-designed components. Currently, these molded parts are 100% manually inspected before delivery to the customer. Workers inspect each part with a magnifying lens to ensure that it contains no defects. These parts have a much larger diameter than, for example, conventional O-rings and therefore do not have a rigid shape. This makes it difficult to automate the inspection of these engineered molded parts.

The images are taken in a controlled research and development environment. The camera only captures a small segment of the part at a time, and a motor continuously rotates it in front of the camera until every segment of the part has been captured (DF1). This generates 135 slightly overlapping images per part, where each image is a grayscale image with a resolution of about 25 mega pixels.

In this way, we capture 324 defective parts containing five different types of defects, resulting in an initial dataset of 43,740 raw images. Figure 3 shows three exemplary defect types. Most parts are only defective at one location, so most of the segments are considered non-defective. This results in a very unbalanced dataset, where only about 1.5% of images contain a defect. Therefore, we apply an undersampling strategy to balance out the class distribution (DF4). This ensures that the model does not overfit the majority class and sees defective and non-defective images at the same frequency. The final dataset consists of 1,280 images.

6.2. Technical Experiments

The main focus of the experiments is to investigate deep learning strategies based on DP3 and DP4. On the one hand, deep learning models can provide

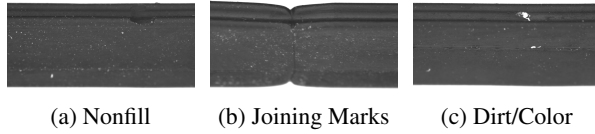


Figure 3: Sample images of different defect types

different amounts of *information value* (DP3) and, on the other hand, they can use different amounts of *knowledge transfer* (DP4). The information value refers to the output of a deep learning model and impacts the explainability and comprehensibility of the model decision. A model can predict whether an image contains a defect or not (binary classification), which defect type it contains (multi-class classification) or where the defect is located in the image (segmentation). However, we hypothesize that an increase in information value is associated with higher learning difficulty, which leads to decreased model performance. Knowledge transfer relates to the amount of abstract knowledge being transferred from other tasks or domains. A model can be trained solely on the data collected for the task, or it can utilize knowledge from a model that has been trained on a generic dataset like ImageNet [49] (generic transfer). Another option is to transfer knowledge from a model that has been trained on an industrial dataset that is supposedly more similar to the target dataset than a generic dataset (industrial transfer). We hypothesize that an increase in knowledge transfer leads to better model performance and shorter training times.

Combining these two dimensions leads to nine different experimental scenarios (Figure 4). We conduct eight experiments in which we cover seven of the scenarios. The two remaining scenarios are not covered due to the lack of an appropriate industrial dataset. The binary classification with industrial transfer scenario is covered by two experiments using two different transfer approaches (DF8 and DF9). However, all experiments also cover the remaining design principles and features from the framework (Table 1).

In the first three experiments (E1, E2, E3), we train binary classification models using different knowledge transfer strategies. The models build on a modified ResNet50 [50] architecture and consist of five blocks of convolutional layers, a global average pooling layer, a dropout layer (DF11), and a fully-connected layer for binary classification (DF5). In experiment E1, we do not transfer any knowledge and initialized the weights randomly. In experiment E2, we apply a generic knowledge transfer by using pre-trained weights from ImageNet [49] (DF8). In experiment E3, we apply an industrial knowledge transfer by using pre-trained

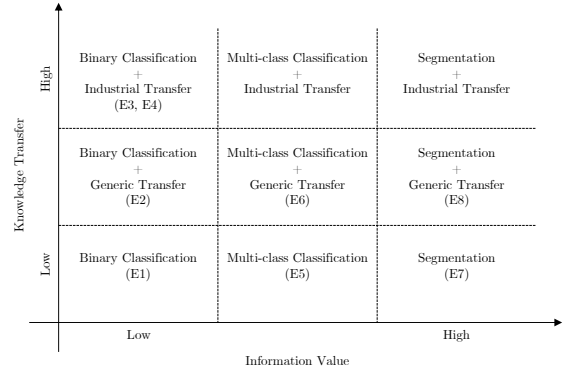


Figure 4: Overview of experimental scenarios addressing information value and knowledge transfer levels

weights from a model trained on a Kaggle dataset [51] (DF8). This dataset consists of 12,586 images of steel defects and contains four types of defects. We chose this dataset because of its relatively large size, visual similarity, and public availability. In experiments E2 and E3, we apply data augmentation by randomly flipping, zooming, and shifting the images (DF12).

In experiment E4, we use a modified CycleGAN [52] to train a network that transforms images of rubber parts into steel images (DF9). We then use this network and a binary classifier trained on the steel dataset to detect rubber part defects.

In experiments E5 and E6, we train multi-class classification models using different amounts of knowledge transfer. We use the same architecture as in the first three experiments, except that we modify the output layer for a multi-class classification with six classes (DF6). In experiment E5, we do not transfer any knowledge and initialize the weights randomly. In experiment E5, we apply a generic knowledge transfer by using pre-trained weights from ImageNet (DF8). In experiment E6, we apply the same data augmentation transformations as in experiments E2 and E3 (DF12).

In experiments E7 and E8, we train segmentation networks using different amounts of knowledge transfer. The output of these networks is a pixel-wise mask of the input image, indicating which pixels belong to which class (DF7). The networks are modified U-Nets [53].

6.3. Evaluation

To better compare the different deep learning strategies, the problem of surface defect detection is translated into a binary classification problem. We evaluate the multi-class classification models in a

Table 1: Overview of the experiments and the implemented design features

Experiment	DF1	DF2	DF3	DF4	DF5	DF6	DF7	DF8	DF9	DF10	DF11	DF12
E1	x	x	x	x	x	-	-	-	-	x	x	-
E2	x	x	x	x	x	-	-	x	-	x	x	x
E3	x	x	x	x	x	-	-	x	-	x	x	x
E4	x	x	x	x	x	-	-	-	x	-	x	x
E5	x	x	x	x	-	x	-	-	-	x	x	-
E6	x	x	x	x	-	x	-	x	-	x	x	x
E7	x	x	x	x	-	-	x	-	-	x	-	-
E8	x	x	x	x	-	-	x	x	-	x	-	-

one-vs-all fashion and transform the segmentation models' output segmentation masks into binary classifications. We use a simple thresholding method, where a segmentation mask is considered a positive classification if the sum of the pixel values of the segmentation mask is above a certain threshold. The threshold value is set to achieve the best possible classification accuracy in each experiment, respectively.

For all experiments, we report accuracy, precision and recall as well as F_1 score as the primary evaluation metric (Table 2). Note that the F_1 score is chosen because it more accurately captures a classifier's performance on unbalanced datasets. This is relevant for the multi-class classification models since the different defect types are represented unevenly.

Table 2: Performance metrics of binary, multi-class classification and segmentation experiments

Experiment	Accuracy	Precision	Recall	F_1 score
E1	0.664	0.666	0.664	0.665
E2	0.974	0.974	0.974	0.974
E3	0.977	0.977	0.977	0.977
E4	0.500	0.250	0.500	0.333
E5	0.549	0.338	0.276	0.28
E6	0.930	0.903	0.894	0.898
E7	0.698	0.702	0.698	0.697
E8	0.930	0.930	0.930	0.930

Considering the binary experiments, E3 (industrial transfer) outperforms the other models with a score of 0.977 in all metrics.

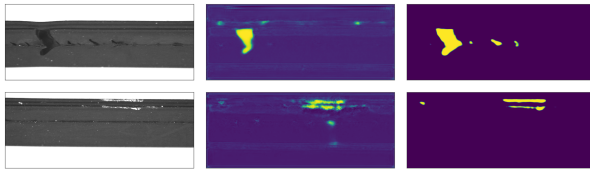


Figure 5: Example inputs (left) and the corresponding predicted segmentation masks in segmentation experiment E7 (middle) and E8 (right)

Overall, binary classification performs slightly better

than multi-class classification as well as segmentation. However, when focusing on the F_1 score, segmentation outperforms multi-class classification.

7. Discussion

By instantiating the proposed design features and their respective design principles through the conducted experiments, applicability and usefulness of the deep learning strategies framework on an industrial surface defect detection tasks is demonstrated. In the following, we separately discuss design principles and their impact.

To the best of our knowledge, this article is the first one applying deep learning methods in the context of surface defect detection of engineered molded parts. The characteristics of these rubber parts present unique challenges to the application of deep learning methods. The very small size of the defects compared to the parts' large surface constitutes a major challenge when capturing the defects with cameras. We address this issue by *segment-wise examination (DP1)*. In particular we capture multiple segments (DF1) for each part and label (DF2) and classify (DF3) the segments separately. This enables us to achieve the required image quality for the detection of surface defects. We also apply an undersampling strategy (DF4) to *balance the dataset (DP2)*.

Furthermore, before conducting the experiments, we hypothesised that an increase in *information value (DP3)* is associated with higher learning difficulty, which leads to decreased model performance. In experiments E2, E6, and E8, we investigate different information values by implementing either a binary classification model (DF5), a multi-class classification model (DF6), or a segmentation model (DF7) with generic knowledge transfers. By comparing the results of these experiments, we see that the model performance decreases when comparing the binary classification (E2) to the multi-class classification model (E6). However, the segmentation model (E8) achieves the same accuracy score as the multi-class classification model and reaches even higher precision, recall, and

F_1 scores. A possible explanation for these results lies in the nature of the task itself. The problem of image segmentation is essentially a problem of image classification on the pixel level. In image segmentation, every pixel of an image is associated with a label and constitutes a training sample to the algorithm. Thus, with the same number of training images, the segmentation model can access more training samples and receives more expert knowledge than the binary or multi-class classification models. The larger amount of training samples or label information might have outweighed higher learning difficulty. For practitioners, this would mean that there is no direct trade-off between model performance and information value and that it could actually be beneficial to increase the information value provided by a model without performance loss.

We also hypothesized that an increase in *knowledge transfer* (DP4) leads to better model performance as well as faster training times. From the results presented, it is clear that the use of pre-trained weights (DF8) impacts model performance and training time. In the binary classification, the multi-class classification, and the segmentation experiments, the models using pre-trained weights (E2, E3, E6, E8) outperform the models without knowledge transfer (E1, E5, E7). Additionally, the models with pre-trained weights converge faster than the models trained from scratch. Figure 6 shows saliency maps [54] of the binary classification models for five exemplary samples. We can see that the model without knowledge transfer (second column) did not learn to detect defects but instead pays attention to some other pattern in the data. The models with knowledge transfer (third and fourth column) learned to recognize defects correctly. The pre-trained weights from a knowledge transfer leverage additional amounts of training data and produce more sensitive gradients than randomly initialized weights. This helps the model to converge towards the global minimum faster. However, the binary classification experiments results suggest that an industrial knowledge transfer (E3) offers only a marginal improvement over a generic knowledge transfer (E2). This might be due to the steel dataset not being similar enough to the rubber part dataset. Even though both datasets contain industrial surface defects, the steel and rubber part images' visual appearance is still quite different. Therefore, the steel dataset might not contain significantly more domain-specific knowledge than the generic ImageNet dataset. From this standpoint, the results can be considered a positive indicator that an actual industrial knowledge transfer, for example, from one molded part type to another one, can produce more significant performance improvements. This

assumption should be addressed in future research. For practitioners, the key finding is that already a generic knowledge transfer leads to significant performance improvements.

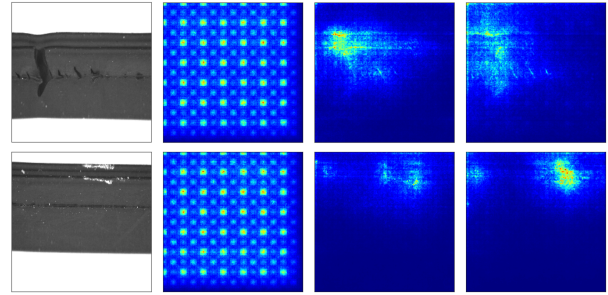


Figure 6: Activation maps of binary classification models E1-E3 (from left to right)

In all experiments, we see that training and validation loss converge together and that the validation loss does not increase again in any of the experiments. Therefore, we conclude that the applied *regularization* (DP5) techniques, such as early stopping (DF10), dropout (DF11), and data augmentation (DF12) features successfully prevent the models from overfitting and address design requirement DR4.

8. Conclusion

This article utilizes a design science research approach to investigate suitable strategies that enable the successful application of deep learning methods in industrial surface defect detection systems. More specifically, we conceptualized a framework of interrelated design requirements, design principles, and design features that captures suitable deep learning strategies for industrial SDDS. In a series of experiments, we utilized the framework to build different deep learning models in an industrial case study. We achieved a 97.7% accuracy in the binary classification of molded part defects using only a very small dataset. The evaluation results showed that transferring knowledge from a generic dataset significantly improves the performance of models for industrial applications. Furthermore, the results indicated that deep learning methods can be successfully applied in surface defect detection systems and that our framework provides a set of practical guidelines for developing visual inspection solutions.

The results, however, should be assessed in light of its limitations. A first limitation relates to the experimental evaluation of deep learning strategies in a single use case. While a quantitative and

broader investigation of deep learning strategies is desirable and encouraged, we want to emphasize that while writing this article, there were no sufficiently large and labeled datasets publicly available for most industrial applications. A second limitation refers to the variety and number of conducted experiments. Our experiments are focused primarily on two aspects of the framework. Conducting further experiments would have enabled us to draw more substantiated conclusions about the remaining aspects of the framework. A third limitation relates to the execution of the experiments. Our hyperparameters are based on pre-tests and state-of-the-art recommendations. Conducting a systematic hyperparameter optimization might have resulted in slightly better model performances; however, our main goal is to evaluate different deep learning strategies and not to achieve the best model performance possible.

Beyond the aforementioned opportunities, there are many other possibilities to extend the work of this article. We encourage scholars to further investigate the impact of information value on model performance and the amount of required training data. Another interesting opportunity lies in the further investigation of industrial knowledge transfers with more suitable datasets.

References

- [1] M. Ferguson, R. Ak, Y.-T. T. Lee, and K. H. Law, "Detection and segmentation of manufacturing defects with convolutional neural networks and transfer learning," *Smart and Sustainable Manufacturing Systems*, vol. 2, no. 1, 2018.
- [2] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, "Segmentation-based deep-learning approach for surface-defect detection," *Journal of Intelligent Manufacturing*, vol. 31, pp. 759–776, May 2019.
- [3] J. Beyerer, F. P. León, and C. Frese, *Machine vision: automated visual inspection: theory, practice and applications*. Springer, 2015.
- [4] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electronic Letters on Computer Vision and Image Analysis*, pp. 1–22, 2008.
- [5] E. Oztemel and S. Gursev, "Literature review of industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, pp. 127–182, July 2018.
- [6] D. Weimer, B. Scholz-Reiter, and M. Shpitalni, "Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection," *CIRP Annals*, vol. 65, no. 1, pp. 417–420, 2016.
- [7] R. Ren, T. Hung, and K. C. Tan, "A generic deep-learning-based approach for automated surface inspection," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 929–940, 2017.
- [8] C. C. Aggarwal, *Neural networks and deep learning*. Springer, 2018.
- [9] P. Zschech, J. Walk, K. Heinrich, and N. Kühl, "A picture is worth a collaboration: Accumulating design knowledge for computer-vision-based hybrid intelligence systems," in *29th European Conference on Information Systems (ECIS 2021)*, June 14 - 16, 2021 - Marrakech, Morocco, 2021.
- [10] D. Soukup and R. Huber-Mörk, "Convolutional neural networks for steel surface defect detection from photometric stereo images," in *International Symposium on Visual Computing*, pp. 668–677, Springer, 2014.
- [11] J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, "Machine learning-based imaging system for surface defect inspection," *International Journal of Precision Engineering and Manufacturing-green Technology*, vol. 3, no. 3, pp. 303–310, 2016.
- [12] X. Tao, D. Zhang, W. Ma, X. Liu, and D. Xu, "Automatic metallic surface defect detection and recognition with convolutional neural networks," *Applied Sciences*, vol. 8, no. 9, p. 1575, 2018.
- [13] H. Di, X. Ke, Z. Peng, and Z. Dongdong, "Surface defect classification of steels with a new semi-supervised learning method," *Optics and Lasers in Engineering*, vol. 117, pp. 40–48, 2019.
- [14] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2016.
- [15] D. Racki, D. Tomazevic, and D. Skocaj, "A compact convolutional neural network for textured surface anomaly detection," in *2018 Winter Conference on Applications of Computer Vision*, IEEE, Mar. 2018.
- [16] S. Gregor, D. Jones, *et al.*, "The anatomy of a design theory," *Association for Information Systems*, 2007.
- [17] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision Support Systems*, vol. 15, no. 4, pp. 251–266, 1995.
- [18] A. R. Hevner, "A three cycle view of design science research," *Scandinavian Journal of Information Systems*, vol. 19, no. 2, p. 4, 2007.
- [19] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [20] R. K. Yin, *Applications of case study research*. sage, 2011.
- [21] M. Chanson, A. Bogner, D. Bilgeri, E. Fleisch, and F. Wortmann, "Blockchain for the IoT: privacy-preserving protection of sensor data," *Journal of the Association for Information Systems*, vol. 20, no. 9, pp. 1274–1309, 2019.
- [22] J. Pries-Heje, R. Baskerville, and J. R. Venable, "Strategies for Design Science Research Evaluation.," in *ECIS*, pp. 255–266, 2008.
- [23] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Quarterly*, vol. 37, no. 2, pp. 337–355, 2013.
- [24] V. Murino, M. Bicego, and I. A. Rossi, "Statistical classification of raw textile defects," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, IEEE, 2004.

- [25] O. Silvén, M. Niskanen, and H. Kauppinen, "Wood inspection with non-supervised clustering," *Machine Vision and Applications*, vol. 13, no. 5-6, pp. 275–285, 2003.
- [26] F. Pernkopf, "Detection of surface defects on raw steel blocks using bayesian network classifiers," *Pattern Analysis and Applications*, vol. 7, no. 3, pp. 333–342, 2004.
- [27] X. Xie and M. Mirmehdi, "TEXEMS: Texture exemplars for defect detection on random textured surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1454–1464, Aug. 2007.
- [28] J. Chen and A. K. Jain, "A structural approach to identify defects in textured images," in *Proceedings of the 1988 IEEE International Conference on Systems, Man, and Cybernetics*, IEEE, 1988.
- [29] B. Mallik-Goswami and A. K. Datta, "Detecting defects in fabric with laser-based morphological image processing," *Textile Research Journal*, vol. 70, no. 9, pp. 758–762, 2000.
- [30] W. Wen and A. Xia, "Verifying edges for visual inspection purposes," *Pattern Recognition Letters*, vol. 20, pp. 315–328, Mar. 1999.
- [31] J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, and J. Song, "Detection of defects in colour texture surfaces," in *MVA*, 1994.
- [32] C.-W. Kim and A. J. Koivo, "Hierarchical classification of surface defects on dusty wood boards," *Pattern Recognition Letters*, vol. 15, pp. 713–721, July 1994.
- [33] R. W. Connors, C. W. Mcmillin, K. Lin, and R. E. Vasquez-Espinosa, "Identifying and locating surface defects in wood: Part of an automated lumber processing system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, pp. 573–583, Nov. 1983.
- [34] M. Niskanen, O. Silvén, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering," in *Proceedings of the scandinavian Conference on image analysis*, pp. 336–342, 2001.
- [35] D. Zhu, R. Pan, W. Gao, and J. Zhang, "Yarn-dyed fabric defect detection based on autocorrelation function and GLCM," *Autex Research Journal*, vol. 15, pp. 226–232, sep 2015.
- [36] F. Ade, N. Lins, and M. Unser, "Comparison of various filter sets for defect detection in textiles," in *International Conference on Pattern Recognition*, vol. 1, pp. 428–431, 1984.
- [37] S. A. H. Ravandi and K. Toriumi, "Fourier transform analysis of plain weave fabric appearance," *Textile Research Journal*, vol. 65, pp. 676–683, Nov. 1995.
- [38] J. Hu, H. Tang, K. C. Tan, and H. Li, "How the brain formulates memory: a spatio-temporal model research frontier," *IEEE Computational Intelligence Magazine*, vol. 11, pp. 56–68, May 2016.
- [39] A. Conci and C. B. Proença, "A fractal image analysis system for fabric inspection based on a box-counting method," *Computer Networks and ISDN Systems*, vol. 30, no. 20-21, pp. 1887–1895, 1998.
- [40] A. F. L. Serafim, "Multiresolution pyramids for segmentation of natural images based on autoregressive models: Application to calf leather classification," in *Proceedings IECON'91: 1991 International Conference on Industrial Electronics, Control and Instrumentation*, pp. 1842–1847, IEEE, 1991.
- [41] F. S. Cohen, Z. Fan, and S. Attali, "Automated inspection of textile fabrics using textural models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 8, pp. 803–808, 1991.
- [42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- [43] J. Masci, U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout, "Steel defect classification with max-pooling convolutional neural networks," in *The 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, June 2012.
- [44] L. Shang, Q. Yang, J. Wang, S. Li, and W. Lei, "Detection of rail surface defects based on cnn image recognition and classification," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 45–51, IEEE, 2018.
- [45] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019.
- [46] S. Faghih-Roohi, S. Hajizadeh, A. Nunez, R. Babuska, and B. D. Schutter, "Deep convolutional neural networks for detection of rail surface defects," in *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2016.
- [47] S. Jung, Y. Tsai, W. Chiu, J.-S. Hu, and C.-T. Sun, "Defect detection on randomly textured surfaces by convolutional neural networks," in *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 1456–1461, IEEE, 2018.
- [48] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.
- [50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [51] Severstal, "Severstal: Steel Defect Detection." <https://www.kaggle.com/c/severstal-steel-defect-detection>, 2019. [Online].
- [52] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.
- [53] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [54] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.